Scalable Linear Solvers

CASC Center for Applied Scientific Computing

The scalable linear solvers project is developing scalable algorithms and software for the solution of large, sparse linear systems of equations on massively parallel computers having upwards of 10,000 processors.

We wish to significantly speed up the solution of the linear systems that arise in many large-scale scientific simulation codes. Applications of interest include radiation diffusion and transport, structural dynamics, flow in porous media, and magnetic fusion energy. The linear systems result from discretizations of partial differential equations on structured, block-structured, and unstructured meshes.

This work is supported by the ASCI (Advanced Simulation Computing Program) and the SciDAC (Scientific Discovery through Advanced Computing) program. These programs focus on developing scalable algorithms and software for solving large, sparse linear systems of equations on parallel computers. The problems of interest arise in the simulations codes being developed to study physical phenomena in the defense, environmental, energy, and biological sciences.

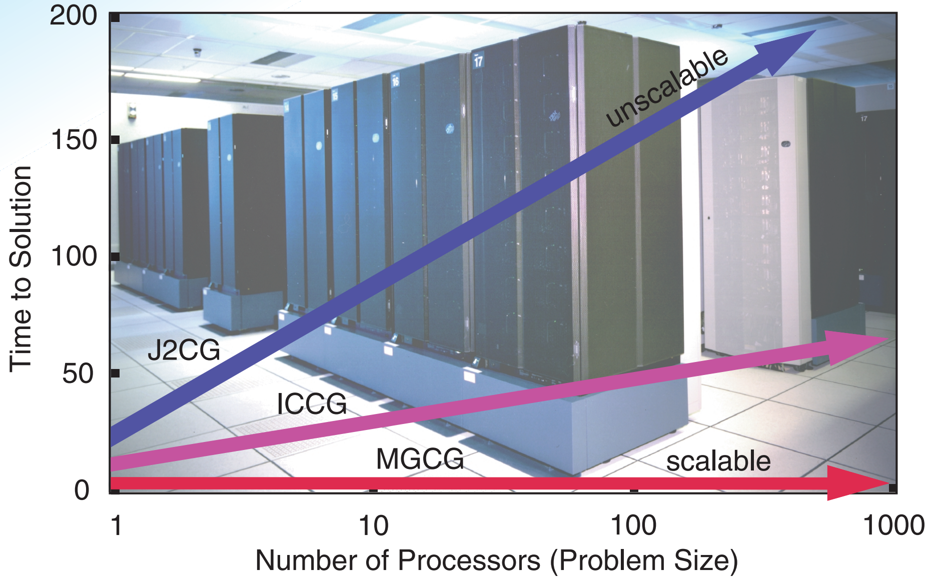


Figure 1. Scalable linear solvers (such as multigrid) enable terascale simulation by keeping solution time constant as the problem size increases with the number of processors. J2CG, ICCG, and MGCG are conjugate gradient algorithms with Jacobi, incomplete Cholesky, and multigrid preconditioners, respectively.

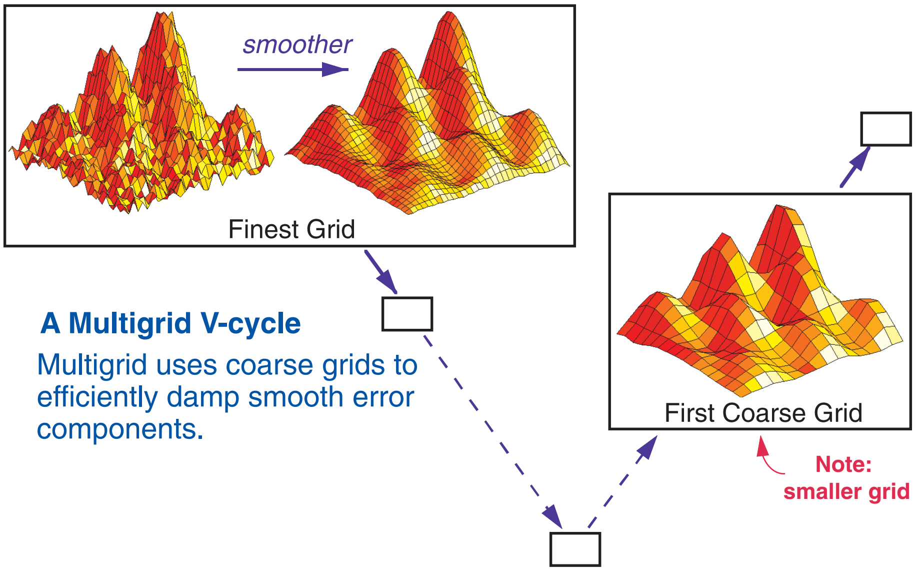
The Need for Scalable Algorithms

Computer simulations play an increasingly important role in scientific investigations, supplementing (and in some cases, supplanting) traditional experiments. In engineering applications, such as automotive crash studies, numerical simulation is much cheaper than experimentation. In other applications, such as global climate change, experiments are impractical (or unwise), and simulations are used to explore the fundamental scientific issues. Finally, in the area of nuclear weapons stockpile stewardship, full-blown experiments are prohibited by the Comprehensive Test Ban Treaty, and detailed numerical simulations are needed to fill the resulting void. Within the DOE and elsewhere, codes are being developed to solve highly resolved three-dimensional problems that require the computational speed and large memory of the massively parallel computers.

Although parallel processing is necessary for the numerical solution of these problems, alone it is not sufficient. The project also requires scalable numerical algorithms. By “scalable” we generally mean the ability to use additional computational resources effectively to solve increasingly larger problems. Many factors contribute to scalability, including the architecture of the parallel computer and the parallel implementation of the algorithm. One important issue, however, is often overlooked: the scalability of the algorithm itself. Here, scalability is a description of how the total computational work requirements grow with problem size, which can be discussed independent of the computing platform.

Many of the algorithms used in today’s simulation codes are based on yesterday’s unscalable technology. This means that the work required to solve increasingly larger problems grows much faster than linearly (the optimal rate). The use of scalable algorithms can decrease simulation times by several orders of magnitude, thus reducing a two-day run on an MPP to 30 minutes (Figure 1). Furthermore, the codes that use this technology are limited only by the size of the machine’s memory, because they are able to effectively exploit additional computer resources to solve huge problems.

Scalable algorithms enable the application scientist to both pose and answer new questions. For example, if a given simulation (with a particular



A Multigrid V-cycle

Multigrid uses coarse grids to efficiently damp smooth error components.

Figure 2. The down-cycle of a multigrid V-cycle uses smoothers to damp oscillatory error components at different grid scales. The up-cycle corrects the smooth error components remaining on each grid level by using the error approximations on coarser (i.e., smaller) grids.

resolution) takes several days to run, and a refined (i.e., more accurate) model would take much longer, the application scientist may forego the larger, higher fidelity simulation. He or she also may be forced to narrow the scope of a parameter study because each run takes too long. By decreasing the execution time, a scalable algorithm allows the scientist to do more simulations at higher resolutions.

Linear Solver Research Directions

In many large-scale scientific simulation codes, the majority of the run time is spent in a linear solver. For this reason, much of the scalable algorithms research and development is aimed at solving these large, sparse linear systems of equations on parallel computers. The scalable linear solver algorithms we develop are implemented and available as part of the *hypre* library.

Multigrid is an example of scalable linear solver technology. It uses a relaxation method like Gauss–Seidel to efficiently damp high-frequency error, leaving only low-frequency, or smooth, error. The multigrid idea is to recognize that this low-frequency error can be accurately and efficiently solved for on a coarser (i.e., smaller) grid. Recursive application of this idea to each consecutive system of coarse-grid equations leads to a multigrid V-cycle (Figure 2). If the components of the V-cycle are defined properly, the result is a method that uniformly damps all error frequencies with a computational cost that depends only linearly on the problem size. In other words, multigrid algorithms are scalable.

For linear systems defined on structured meshes (e.g., logically rectangular meshes) and semi-structured meshes (e.g., locally refined meshes), we are developing geometric multigrid methods. An algorithm of this type was used in a three-dimensional parallel groundwater simulation (using eight million spatial zones) to speed up the linear solves by a factor of 120 with nearly 90% scaled efficiency on 256 processors of the Cray T3D. We have developed two geometric multigrid algorithms (SMG and PFMG), which are available in the *hypre* library. Parallel structured multigrid codes are now being run on ASCI platforms and used in ASCI codes. For example, the solution of a one-billion-unknown system took less than 90 seconds using 3,150 processors.

For linear systems defined on unstructured meshes, it is difficult to use geometric information in a way that is simple, straightforward, and portable from application to application. For this reason, we are developing new algebraic multigrid methods. This type of method has been used successfully to solve problems in a large number of application areas, but there are still important problems that cannot be solved effectively using current techniques. Our research focuses on efficient algebraic multigrid solvers for finite element applications, and effective parallelization strategies for large-scale algebraic multigrid methods. The algebraic multigrid code, Boomer AMG, available in the *hypre* library, has successfully solved linear systems using over a thousand processors of ASCI Blue at LLNL.

To enhance robustness, we often use multigrid as a preconditioner for Krylov methods such as conjugate gradients. In general, preconditioners can be designed to have a multilevel structure and give scalable performance like multigrid. Our research includes the development of multilevel versions of approximate inverse and incomplete factorization (ILU) techniques. These techniques are algebraic in nature and work on unstructured meshes. Results show that they scale almost as well as multigrid methods and are often more robust across a broader class of problems.

For more information about Scalable Linear Solvers, contact Robert Falgout, (925) 422-4377, rfalgout@llnl.gov; or Jim E. Jones, (925) 423-5194, jjones@llnl.gov.

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.